SEU System Analysis: Not Just the Sum of All Parts

Melanie Berg, AS&D Inc. in support of NASA/GSFC

Melanie.D.Berg@NASA.gov

Kenneth Label: NASA/GSFC

List of Acronyms

- **Analog-to-Digital Converter (ADC)**
- **Application specific integrated circuit (ASIC)**
- **Block random access memory (BRAM)**
- **Combinatorial logic (CL)**
- **Device Under Test (DUT)**
- **Digital clock manager (DCM)**
- **Digital signal processor (DSP)**
- **Edge-triggered flip-flop (DFF)**
- **Error rate (dE/dt)**
- **Field programmable gate array (FPGA)**
- **Linear energy transfer (LET)**
- **Localized triple modular redundancy (LTMR)**
- **Look up table (LUT)**
- **Single event effects (SEEs)**
- **Single event functional interrupt (SEFI)**
- **Single event transient (SET)**
- **Single event upset (SEU)**
- **Single event upset cross section (σ_{SEU})**
- **Static random access memory (SRAM)**
- **System frequency (*fs*)**
- **Triple modular redundancy (TMR)**
- **Windowed shift register (WSR)**

2

To be presented by Melanie D. Berg at the Single Event Effects (SEE) Symposium and the Military and Aerospace Programmable Logic Devices (MAPLD) Workshop, La Jolla, CA, May 19-22, 2014.

Acknowledgements

- Defense Threat Reduction Agency (DTRA)
- NASA Electronic Parts and Packaging (NEPP)
- Radiation Effects and Analysis Group (REAG) led by Kenneth LaBel and Jonathan Pellish.

3

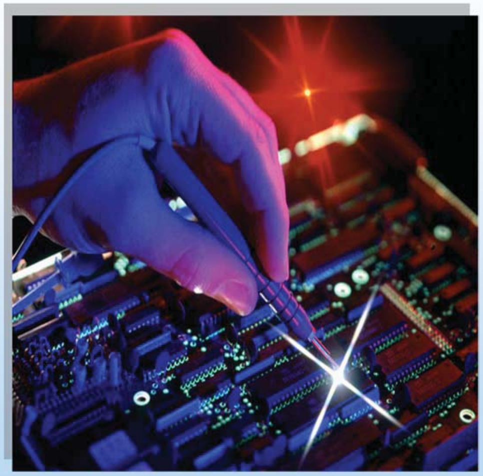
Motivation

- SEU analysis of a system is complex.
- Currently, system SEU analysis is performed by component level partitioning and then:
  - Use the most dominant σ_{SEU}s for system error rate calculations, or
  - Sum component σ_{SEU}s for system error rate calculations.
- In many cases, system error rates are overestimated.
- Overestimation can cause overdesign:
  - Cost, schedule, functionality, and validation/verification can be compromised.
- The scope of this presentation is to discuss the risks involved with our current method of SEU analysis for complex systems.

4

Scope of Systems Regarding This Presentation

- Board or box level group of components:
  - FPGA, ASIC, ADC, microprocessor, microcontroller, memory, oscillator, voltage regulator, operational amplifier, etc…,
- Network of components within a digital design implemented in an ASIC or FPGA
  - DFFs, combinatorial logic, clock managers (DCMs), look up tables (LUTs), etc…,



5

Complex System SEU Evaluation

- Challenges of evaluating complex systems:
  - Fitting the entire system in an accelerated beam,
  - Having the entire system accessible for testing,
  - Enhancing the visibility of SEU-induced system errors,
  - Controlling and monitoring the system during accelerated testing, and
  - Performing SEU data analysis.
- Hence, SEU testing is generally performed using system partitions.
  - Partitioned component co-dependencies within the system should be determined and taken into account when performing SEU analysis.
  - Generally, there should not be just one SEU error rate for a system. Completely independent applications should have unique SEU error rates calculated

6

Component Level Error Rates versus Error Responses

- SEU error rates: How often a component reaches an erroneous-state due to induced noise from ionization (SET or SEU).
- SEU error response: What happens when a component incurs an SET or SEU.
- Component Error rates are generally obtained from accelerated testing and σ_{SEU} extrapolation.
- Other fault injection techniques exist, however, they are generally used for error-response studies.

7

Several Factors That Are Generally Not Taken Into Account during Component Level SEU Testing

- How often is the component used in the system?
- Is the component masked?
- Will the system be affected if the component incurs an SEU?
  - Can the SET dissipate prior to causing a system error?
  - Will the SET or SEU be captured by the system?
  - Is the SEU masked or is the system not communicating with the component while the SEU exists?
- If several of the same components exist, are they all equally likely to cause a system upset?
  - Can the analysis be considered linear, i.e., can we sum the component SEU error rates?

8

When Dominant Component Error Rates Can Be Used as the System Error Rate

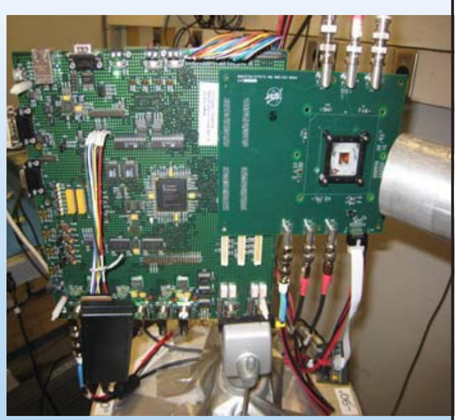
- The easiest system to evaluate is one where a dominant component error rate can be applied.
  - For example, a design implemented in a commercial SRAM-based FPGA. The configuration upset rates dominate all others.
- However, this is not always straightforward:
  - If components are SEU tested separately, co-dependencies are not taken into account. This can change error rates significantly.
  - If components are co-dependent, it is important to either test as a system (sub-system) or evaluate how the co-dependencies can affect error rates.
    - For example, testing DFFs test structures versus DFFs in a system design.

9

Characterizing SEUs: Radiation Testing and SEU Cross Sections

SEU Cross Sections (σ_{seu}) characterize how many upsets will occur based on the number of ionizing particles the device is exposed to

$$\sigma_{seu} = \frac{\#errors}{fluence}$$



Terminology:

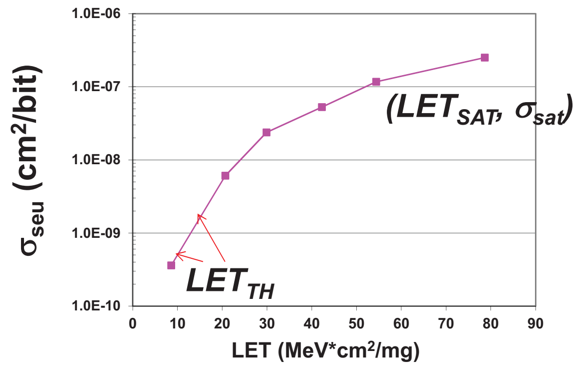
- Flux: Particles/(s·cm^2)
- Fluence: Particles/cm^2
- σ_{seu} is calculated at several LET values (particle spectrum)

10

Characterizing SEUs: LET vs. SEU Cross Section Graph and How They Relate to Error Rates

$$\sigma_{seu} = \frac{\#errors}{fluence}$$

dE/dt is calculated by integrating σ_{SEU} over the LET spectrum using a Weibull fit



LET_{SAT} = Saturated LET

LET_{TH} = Threshold LET

σ_{SAT} = Saturated SEU Cross Section

GEO Upset Rate:

$$\frac{dE}{dt} \approx \frac{C * \sigma_{sat}}{LET_{0.25}^{\;2}}$$

After Ed Petterson's figure of merit

C varies based on the orbit. For GEO, values between 200 and 400 are common.

11

Example of Dominant σ_{SEU}

- **If the co-dependency between components is insignificant, then component error-rates can be summed; e.g, FPGA high-level internal structures:**

SEU Cross-Sections (σ_{SEU}) = #upsets/particle/cm²

$$P(fs)_{error} \propto P_{Configuration} + P(fs)_{functionalLogic} + P_{SEFI}$$

Design σ_{SEU} ***Configuration σ_{SEU}*** ***Functional logic*** ***SEFI σ_{SEU}***

Sequential and Combinatorial logic (CL) in data path

Global Routes and Hidden Logic

With hardened configuration and hardened global routes (e.g., Microsemi RTAX2000s)

12

To be presented by Melanie D. Berg at the Single Event Effects (SEE) Symposium and the Military and Aerospace Programmable Logic Devices (MAPLD) Workshop, La Jolla, CA, May 19-22, 2014.

Taking into Account The Non-Linearity of Systems during the Extrapolation Process

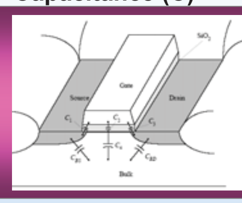
How do we extrapolate σ_{SEU}s to complex designs?

What Forces Non-Linear σ_{SEU} Extrapolation

- **System Block SEUs**
  - How often is the component active?
  - Is the component masked?
  - Are global route SETs taken into account?
- **SETs**
  - Dissipation during propagation
  - Elongation during propagation
  - Masking via logic components
  - Ringing/oscillation due to metastabiity (e.g., transistor push-pull during transient creation or clock tree SETs).

- Cutoff frequency (f_c)
- Resistance (R)
- Capacitance (C)

$$f_c = \frac{1}{2\pi RC}$$

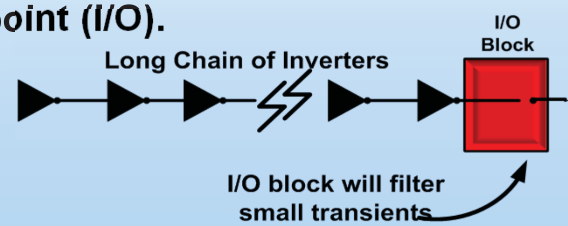


Each capacitance has its own f_c

To be presented by Melanie D. Berg at the Single Event Effects (SEE) Symposium and the Military and Aerospace Programmable Logic Devices (MAPLD) Workshop, La Jolla, CA, May 19-22, 2014.

SET Characterization via Long Inverter Chains

- Common method for testing SET behavior is to use a long chain of inverters.
- Inverter SET cross sections are calculated by counting the number of SETs and dividing by the number of inverters.
- Problem: This method assumes all inverters have the same probability of upset as seen from the observation point (I/O).
- In addition, this method assumes linear behavior.



15

SEU Cross Sections and Error Rates – How We Apply Them to FPGA Designs

- A goal of SEU testing is to provide error rate (*dE(fs)/dt*) predictions to critical missions.
- σ_{SEU}s from SEU testing are used to calculate (*dE(fs)/dt*) .
- $dE(f_s)/dt$ for FPGA and ASIC devices are calculated using:

System upset rate — SEU bit upset — Number of used flip-flops DFFs

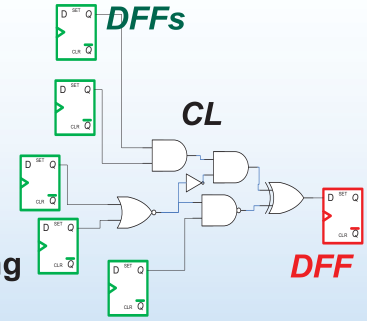
$$\frac{dE(fs)}{dt} < \frac{dE_{bit}(fs)}{dt} * (\#UsedDFFs)$$

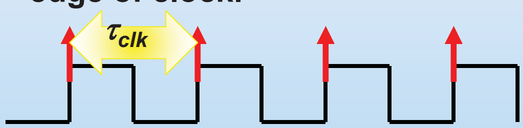
- Assumes linearity – all DFFs are used every cycle and that they have the same probability of upset.

16

Background: Synchronous Design Data Path – Sample and Hold

- **Synchronous design components:**
  - Edge Triggered Flip-Flops (DFFs),
  - Clocks and resets (global routes), and
  - Combinatorial Logic (CL).
- **All DFFs are connected to a clock.**
- **DFFs sample their input at the rising edge of clock.**





Clock Period $\tau_{clk} = \dfrac{1}{fs}$ Frequency

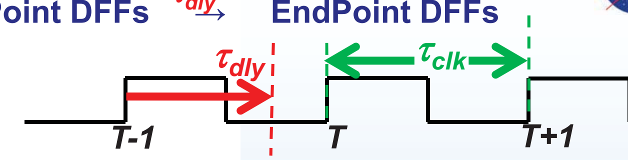
- **CL compute between clock edges.**

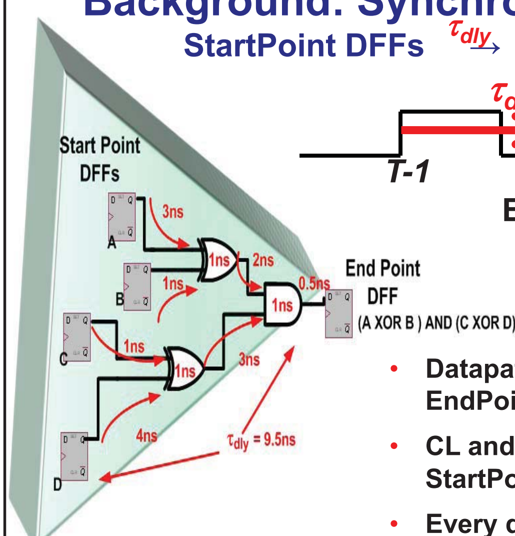
Designs are complex – We modularize for simplicity

17

Background: Synchronous Data Paths:

StartPoint DFFs $\xrightarrow{\tau_{dly}}$ EndPoint DFFs





Every DFF has a function that determines its state

$$EndPoint(T) = f(StartPoints(T-1), CL)$$

- Datapath defined as StartPoint via CL to EndPoint.
- CL and routes create delay (τ_{dly}) from StartPoints to EndPoints.
- Every data path has a unique τ_{dly}.
- τ_{dly} is calculated using Static Timing Analysis (STA) design tools.

Modularization: Every DFF has a unique cone of logic

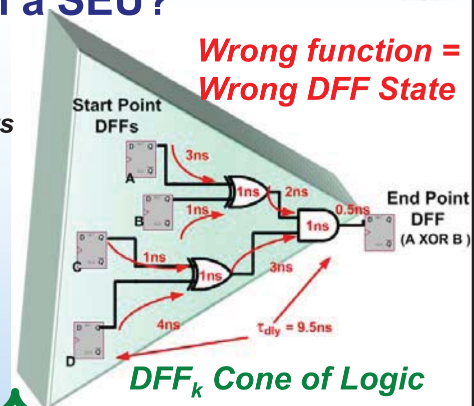
18

How can a DFF Contain an Incorrect State from a SEU?

- *DFFs have various modes of reaching a bad state due to SEUs.*
- *Attribute some modes to EndPoints and some to StartPoints.*

We make a clear distinction between DFF SEUs based on Clock state and Capture.

Wrong function = Wrong DFF State



$\forall_{EndPoint\ DFF}$ ***EndPoint DFF SEUs + StartPoint DFF SEUs + CL SETs***

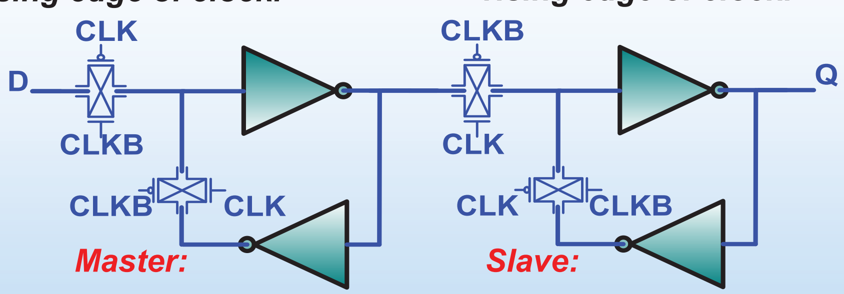
EndPoint DFF SEUs	StartPoint DFF SEUs	CL SETs
DFF upsets that occur at the clock edge.	*DFF upsets that occur between clock edges and are captured by EndPoints.*	*Single Event Transients captured by EndPoints.*

19

Edge Triggered DFFs... Creating Deterministic Boundary Points

D input must be settled by rising edge of clock.

Output will only change at rising edge of clock.



Master:
Clock Low: Transparent
Clock High: Hold

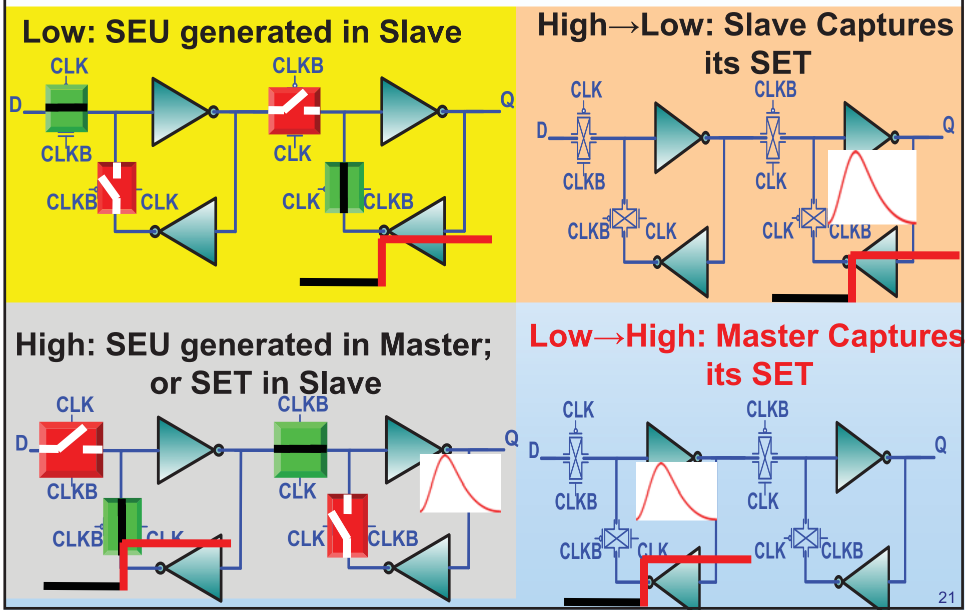
Slave:
Clock Low: Hold
Clock High: Transparent

CLK = clock CLKB = inverted clock

In order to create precise boundary points of state capture, latches are NOT allowed in synchronous designs.

20

StartPoint and EndPoint DFF SEUs as a Function of Clock State ($P(fs)_{DFFSEU}$)



21

Summary of Internal DFF SEUs

$$P(fs)_{DFFSEU} = \alpha P(fs)_{DFFSEU} + \beta P(fs)_{DFFSEU}$$

Percentage of SEUs that occur at rising clock edge

- Master SET gets trapped during transition from transparent to hold state (rising edge of clock).
- This is considered a state change.

EndPoint SEU

Percentage of SEUs that occur between clock edges

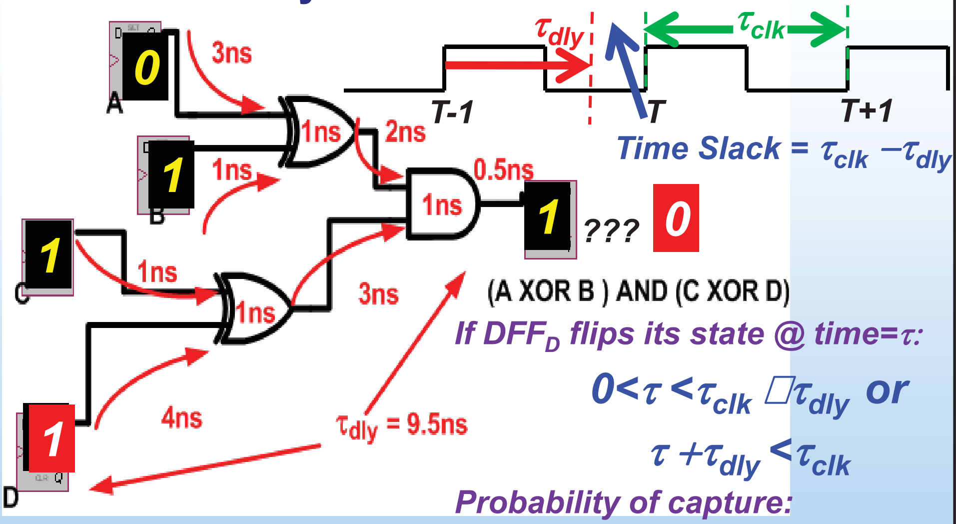
- Master or slave is in hold state or Slave captures its own SET during transition from transparent to hold state.
- This is not considered a definitive state change.
- Must be captured by an EndPoint to cause an incorrect change in system state.

StartPoint SEU

By definition, EndPoint SEUs are already captured into the system. How do StartPoints get captured?

22

How Does a StartPoint SEU get Captured by an EndPoint?



Time Slack = $\tau_{clk} - \tau_{dly}$

(A XOR B) AND (C XOR D)

If DFF_D flips its state @ time=τ:

$0<\tau<\tau_{clk} - \tau_{dly}$ or

$\tau + \tau_{dly} < \tau_{clk}$

Probability of capture:

$1-(\tau_{dly}/\tau_{clk}) = 1-\tau_{dly} fs$

23

Details of Capturing StartPoint DFFs

$$\forall_{DFF}\left(\left(\sum_{j=1}^{\#StartPoint\,DFFs} \beta P(fs)_{DFFSEU(j)} (1-\tau_{dly(j)} fs) P_{logic(j)} \right) \right)$$

Upset generated internally to DFF between clock edges

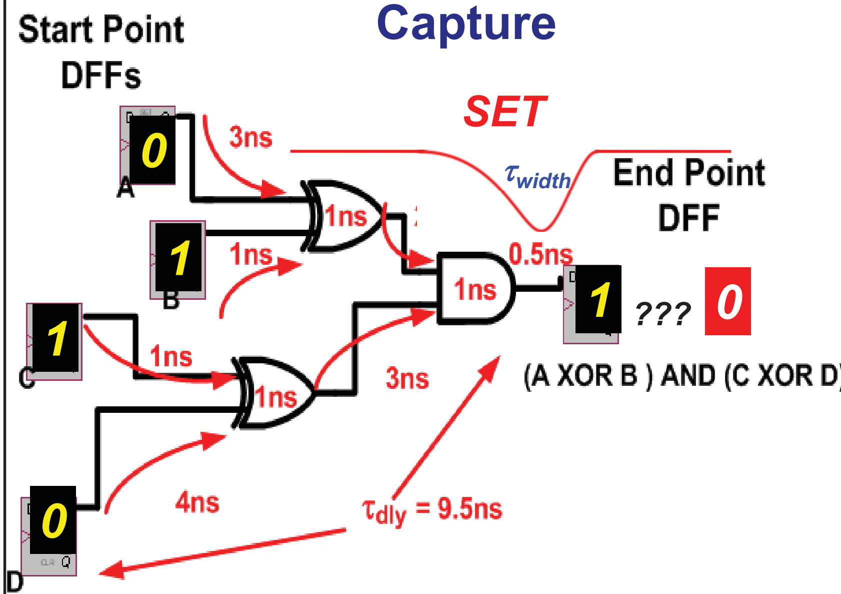
Design Topology and Temporal Masking

Design Topology and Logic Masking

- SEU generation occurs in a StartPoint between rising clock edges ($\beta P(fs)_{DFFSEU}$).
- StartPoint upsets can be logically masked by logic between the StartPoint and its EndPoint.
- Design topology and temporal effects:
  - Increase path delay (# of gates) – decrease probability of capture.
  - Increase frequency – decrease probability of capture.

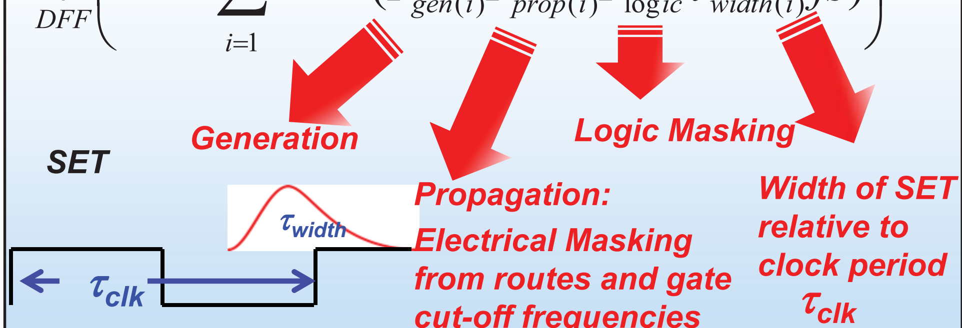
24

Synchronous System: CL SET Capture



Details of CL SET Capture

$$\forall_{DFF}\left(\sum_{i=1}^{\#Combinator\ ialCells}(P_{gen(i)}P_{prop(i)}P_{logic}\tau_{width(i)}fs)\right)$$



- SET Generation (P_{gen}) occurs between clock edges.
- EndPoint DFF captures the SET at a clock edge.
  - Increase frequency – increase probability of capture.
  - Increase CL – increase probability of capture.

Putting it All Together – Analyzed Per Particle Linear Energy Transfer (LET)

EndPoint Logic Masking | EndPoint | StartPoints | CL

$$\sum_{k=1}^{\#EndPoint\ DFFs} P_{logic(k)} * \left(\begin{array}{l} \alpha P(fs)_{DFFSEU(k)} + \\ \sum_{j=1}^{\#StartPoint\ DFFs} \left(\beta P(fs)_{DFFSEU(j)} (1 - \tau_{dly(j)} fs)) * P_{logic(j)} \right) + \\ \sum_{i=1}^{\#CL} (P_{gen_{(i)}} * P_{prop_{(i)}} * P_{logic_{(i)}} * \tau_{width_{(i)}} fs) \end{array} \right)$$

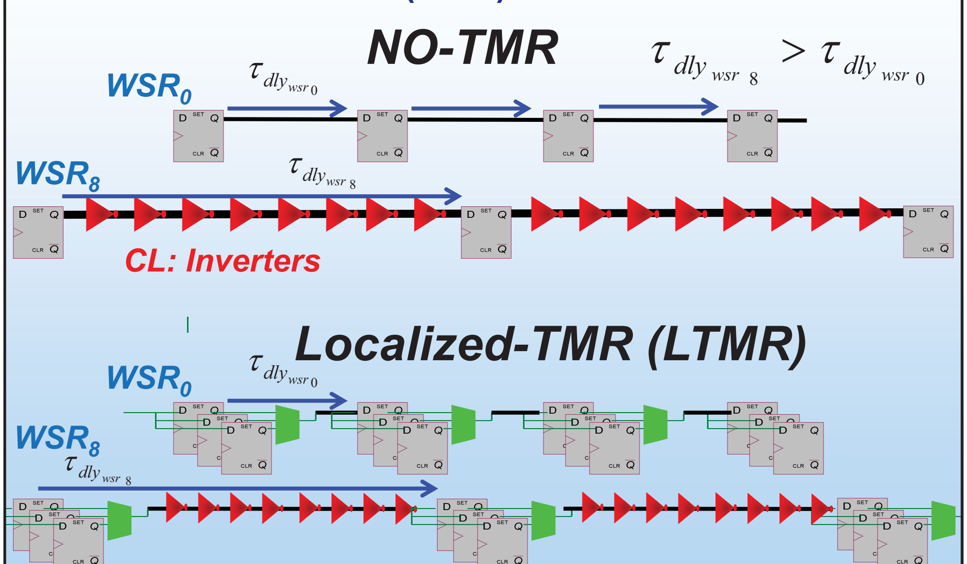
StartPoints and CL need to be captured by an EndPoint… hence data path derating factors exist.

Component Contribution to σ_{SEU} across Frequency and Gate Count

	Frequency	# of Gates in Path
EndPoint	Directly Proportional	N/A
StartPoint	Inversely Proportional	Inversely Proportional
CL	Directly Proportional	Directly Proportional

27

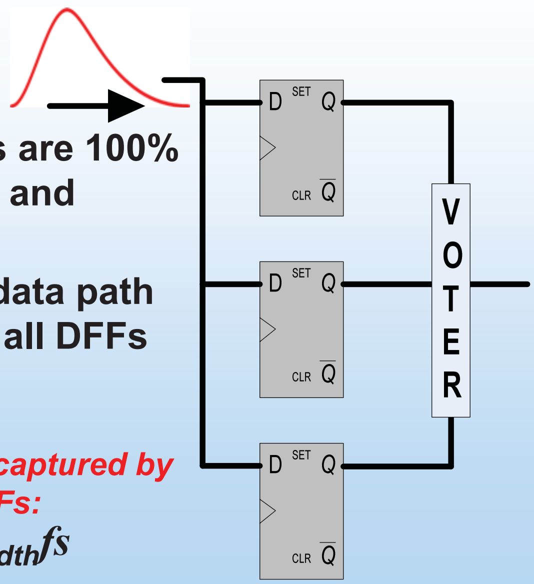
Radiation Test Structures: Windowed Shift Registers (WSR) and Triple Modular Redundancy (TMR)



28

LTMR SEU Response

- Internal DFF upsets are 100% masked: StartPoint and EndPoint P_{logic} = 0;
- SETs from shared data path can propagate into all DFFs
- Voters can upset

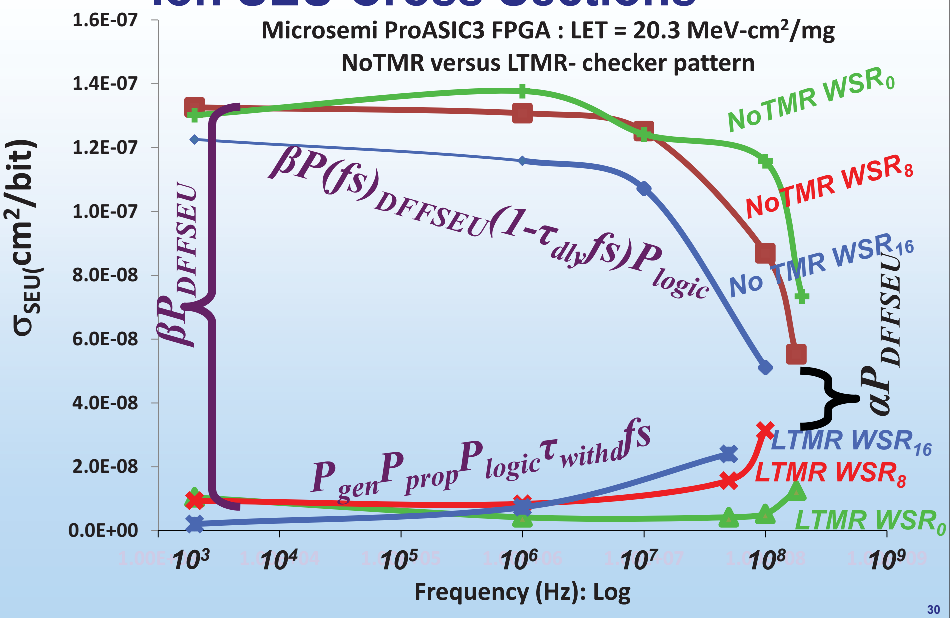


Data path SETs can be captured by the LTMR'd DFFs:

$$P_{gen}P_{prop}P_{logic}\tau_{width}fs$$

29

Using the Model to Analyze Heavy Ion SEU Cross Sections



30

SEU Characterization of A Complex System: Microprocessor

Test-As-You-Fly versus Using Fest Structures and Extrapolation

31

Test Structures versus Final Designs

- Although error rates and error responses are design dependent, useful information can be extrapolated from test structures versus the final design.
- Why use test structures versus final designs?
  - By the time the final design is complete, it is usually too late to perform radiation testing on it.
  - Can be too difficult to apply input-stimuli to a final design.
  - Can be too difficult to monitor DUT responses.

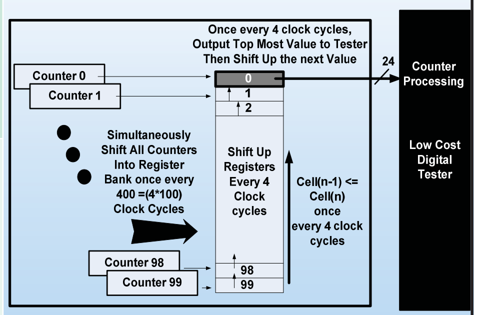
The following slides give more insight into the benefits of using test structures versus full designs during radiation testing.

32

Best Practice for Radiation Testing: Logic Replication for Statistics

Best-Practice for DUT Test Structure Development	How Application-Specific Test Structures Violate Best-Practice Considerations
Test structures should contain a large number of replicated logic in order to increase statistics: e.g., shift-registers with thousands of stages.	• Statistics are poor because usually there is not a significant amount of replication. • In addition, trends for specific elements are not able to be clearly identified / established.

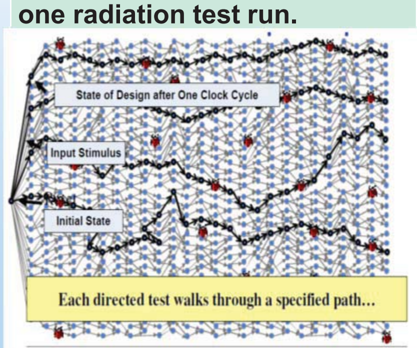
SEU testing with hundreds of counters versus only one



33

Best Practice for Radiation Testing: State Space Traversal

Best-Practice for DUT Test Structure Development	How Application-Specific Test Structures Violate Best-Practice Considerations
A test structure's state space should be traversable such that it can be covered within one radiation test run.	The state space of a complex design cannot be traversed within one radiation test run.
	Hence, a significant amount of circuitry and system states are not tested.
	The result is SEU data that are uncharacteristic of the design.



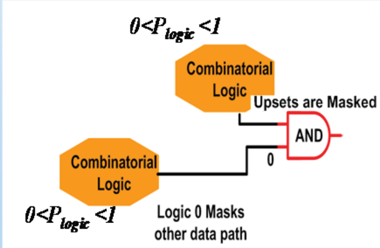
34

Best Practice for Radiation Testing: Logic Masking

Best-Practice for DUT Test Structure Development	How Application-Specific Test Structures Violate Best-Practice Considerations
Logic masking should be minimized or controllable.	Application-specific test structures contain a significantly higher number of masked data paths than test structures.

P_{logic} is the probability that an upset will be masked from being captured by the system.

$P_{logic} = 0$: path is 100% masked
$P_{logic} = 1$: path has no masking

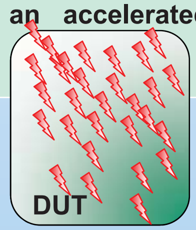


35

Best Practice for Radiation Testing: Avoiding Unrealistic SEU Accumulation

Best Practice characteristics of a DUT design	How Application-Specific Test Structures Violate Best-Practice Considerations
Avoid unrealistic SEU accumulation from accelerated testing: • Flush through test structures; e.g., shift-registers. • Small number of gates per sub-test structure; e.g., testing hundreds of counters.	Application-specific test structures take up most of the DUT's area. There are a lot of co-dependencies between logic. Hence, it is difficult to control SEU accumulation in an accelerated test environment.

SRAM Based FPGAs: Scrubbing (correcting) configuration SEUs. Extremely important during accelerated testing… must keep up with the particle flux to avoid accumulation



36

Best Practice for Radiation Testing: Increasing Visibility

Best Practice characteristics of a DUT design	How Application-Specific Test Structures Violate Best-Practice Considerations
All (or a significant percentage of) potential upsets should be observable during testing.	A significant number of upsets in a complex design are generally not observable during radiation testing.
Test structures can easily be designed to enhance observable nodes; e.g., shift-registers and counters.	This is true mostly because of logic masking, limitations in state space traversal, limitations in I/O count, or time of upset propagation to observable node.

37

Benefits of Testing Application Specific Designs

- Increase observation error responses specific to the application.
- However, the user must be aware of the following:
  - Unrealistic SEU accumulation in an accelerated environment.
  - Limited visibility due to masking and fractional state space traversal.
  - Poor statistics due to the variance in design circuits.
- σ_{SEU}s will most likely have a large variance if circuits are not able to be isolated and controlled.

38

CASE Study

- DUT is a Xilinx V5QV – radiation hardened FPGA.
- Application-specific test structure is an embedded microprocessor (Micro-blaze™).
- Goal is to determine error rates for using an embedded Micro-blaze™ processor in the Xilinx V5QV with and without cache.
  - Question: Does using cache in embedded memory increase the σ_{SEU}s such that the Micro-blaze™ will not meet project requirements?

39

Suggestions on How to Test the Application Specific Design

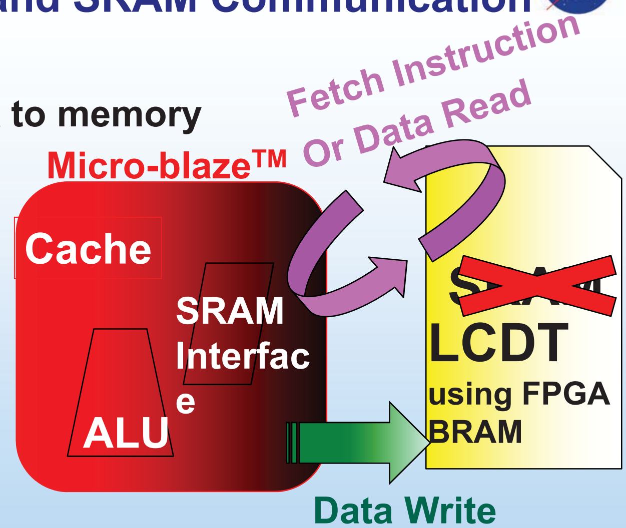
- Because the goal is to study caching SEU effects, test-plan should have a test design that contains cache and one that does not.
- Test basic structures such as shift-registers and counters to get an underlying understanding of device SEU characteristics.
- Basic test-structure analysis characterizes:
  - Sequential memory elements (DFFs),
  - Combinatorial logic (CL), and
  - Global routes.
- Increase visibility of the Micro-blaze™ during testing.

40

Processor and SRAM Communication

SRAM: Static random access memory
BRAM: Block random access memory

- Processors talk to memory
  - Most processor radiation tests detect errors by erroneous SRAM memory writes.
  - Visibility is significantly limited.

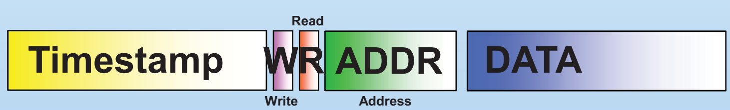


- *We increase visibility by replacing external SRAM with the REAG low-cost digital Tester (LCDT)*

41

More on Increasing Visibility with Microprocessor Testing (1)

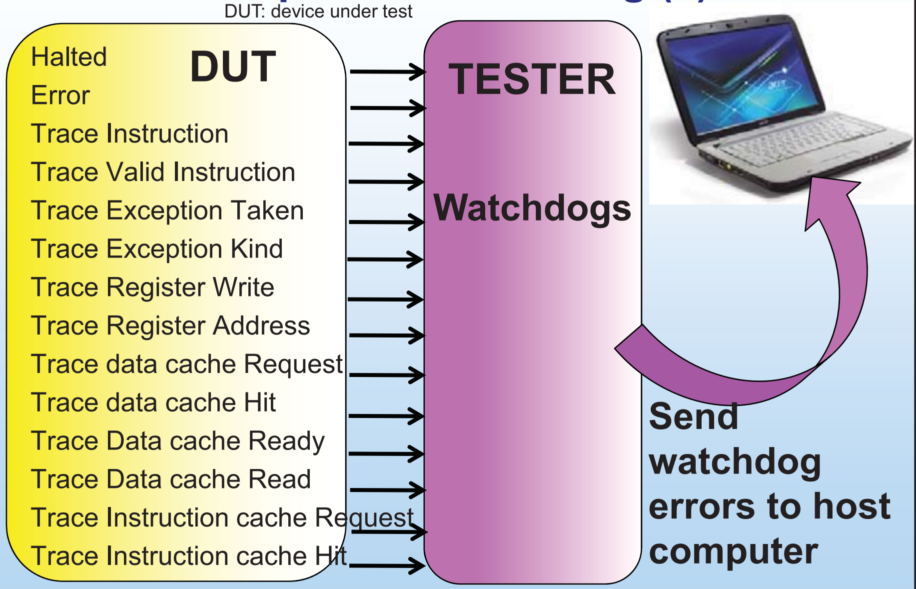
- As previously stated, the embedded SRAM in the tester (BRAM) takes the place of normal memory accesses.
- In addition, each memory access is time stamped and logged in alternate bank of BRAM. Only the last 512 accesses are kept.
- After each test run, the time stamped logs are output to the user.



42

More on Increasing Visibility with Microprocessor Testing (2)

DUT: device under test



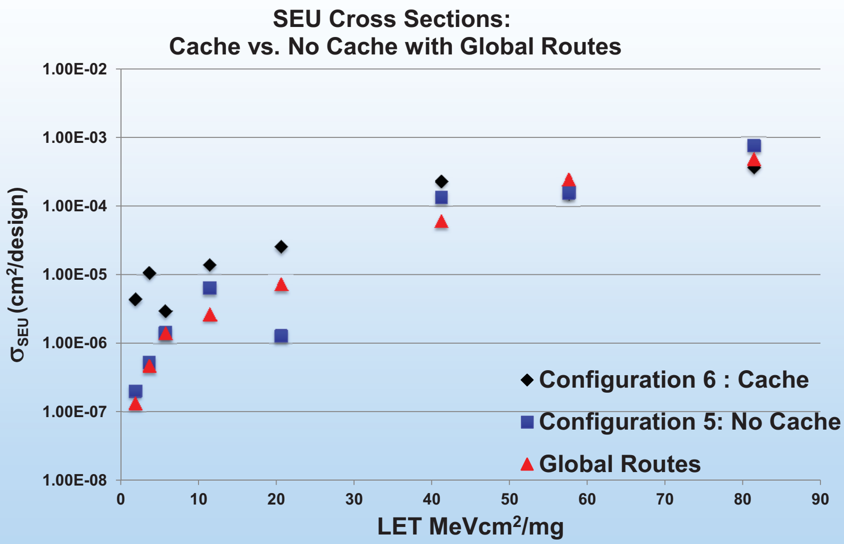
43

Summary of Case Study Test Enhancements

- **Visibility was increased by isolating memory accesses as follows:**
  - **Moving the instruction and data storage to the LCDT for traffic observation.**
  - **Performing tests with and without cache to determine the influence cache has on upsets.**
- **Differentiating global upsets from the normal data set:**
  - **Helped to understand which upsets are prominent.**
  - **Gave insight to how the use of cache will affect σ_{SEU}s.**
- **Monitoring internal Micro-blaze™ signals**
  - **σ_{SEU}s are not reliant on detecting erroneous memory read and writes anymore. Data are too limited and uninformative with solely relying on memory reads and writes.**
  - **Can now determine when a processor crashes and how.**

44

Comparing Micro-blazeTM σ_{SEU}s and Global Clock σ_{SEU}s



45

Floor Is Open To Discussion

46